Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432


A INFLUÊNCIA DA ENTROPIA DAS SENHAS NA DEFESA CONTRA ATAQUES DE FORÇA BRUTA

THE INFLUENCE OF PASSWORD ENTROPY IN DEFENSE AGAINST BRUTE FORCE ATTACKS

Eduardo Brandt Silveira da Silva
Graduando em Ciência da Computação no Instituto Federal Catarinense Campus Blumenau
eduardobrandt61@gmail.com

Henrique Heiderscheidt
Graduando em Ciência da Computação no Instituto Federal Catarinense Campus Blumenau
henrique.heid@gmail.com

Luiz Ricardo Uriarte
Doutorado em Engenharia de Produção pela Universidade Federal de Santa Catarina
luiz.uriarte@ifc.edu.br

Resumo:
Senhas são comuns na atualidade e servem para proteger informações sensíveis e pessoais, então é necessária a criação de senhas seguras que não são prontamente descobertas. Para saber se a senha realmente apresenta segurança, existem métricas que calculam a sua defesa contra ataques cibernéticos, além da sua dificuldade de ser identificadas por invasores com intenções criminosas. A entropia define a força de uma senha baseada em sua aleatoriedade, levando em conta a quantidade de caracteres possíveis e a disposição deles em determinado tamanho de senha. Levando em conta a entropia das senhas, a pesquisa consiste em testes utilizando algoritmo de força bruta que testa as possíveis senhas e tenta encontrar a senha atacada, que está transformada por uma função hash. Observou-se que o fator que mais protege contra ataques é a quantidade de caracteres que ela possui, não possuindo padrões comuns em senhas e a função hash utilizada no seu armazenamento, além de não estar em nenhum vazamento de senhas. Dessa forma uma palavra-chave que possui o mesmo número de entropia que outra pode ser quebrada mais velozmente se possuir padrões ou já estiver disponível em vazamentos de dados.
Palavras-chave: *Senha; Segurança; Hash; Entropia; Ataques de força bruta*.

Abstract:
Passwords are common today and serve to protect sensitive and personal information, so it is necessary to create secure passwords that are not readily discovered. To find out if the password is really secure, there are metrics that calculate its defense against cyber attacks, in addition to its difficulty in being identified by attackers with criminal intentions. Entropy defines the strength of a password based on its randomness, taking into account the number of possible

Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432

characters and their arrangement in a given password length. Taking into account the entropy of the passwords, the research consists of tests using a brute force algorithm that tests the possible passwords and tries to find the attacked password, which is transformed by a hash function. It was observed that the factor that most protects against attacks is the number of characters it has, not having common patterns in passwords and the hash function used in its storage, in addition to not being in any password leak. This way a keyword that has the same entropy number as another can be cracked faster if it has patterns or is already available in data leaks.
Keywords: *Password; Security; Hash. Entropy; Brute force attacks.*

1. INTRODUÇÃO

Ao longo do tempo as senhas sempre estão associadas a segurança, de tal forma que o seu uso tenha os mais diversos propósitos. Porém, diferente de uma chave física, mais de uma pessoa pode ter posse da senha de modo a tornar impossível saber se isso realmente ocorre (Burnett, 2006).

Para determinar a força de uma senha, alguns autores utilizam a noção de entropia, que segundo o NIST (National Institute of Standards and Technology), é a quantidade de incerteza que um atacante se depara ao tentar quebrar alguma senha, medida em bits. Para calcular a entropia de uma senha, são utilizadas diversas variáveis e dados, dependendo do método utilizado.

Existem diversas ameaças como por exemplo: algoritmos de força bruta, onde o computador percorre uma lista com todas as possibilidades de caracteres até encontrar a correta, com o tempo dependendo da variedade de caracteres, processamento do computador e também o tamanho da senha; invasão do computador ou monitoramento da rede (*sniffing*), *keyloggers* ou *screenloggers*. Além dos métodos citados anteriormente, existe a engenharia social, que consiste em se aproximar do alvo, fazendo perguntas que parecem inofensivas, mas que levam à descoberta da senha (Kissell, 2018).

Combinado ao método de força bruta ainda existem os ataques de dicionários que são baseados no fato de que muitas pessoas seguem um padrão ao criarem suas senhas, e então acabam criando-as utilizando palavras simples e coisas de seu cotidiano. Esses dicionários são gerados, por exemplo, através de listas de senhas mais utilizadas na internet, uma relação de pessoas, cidades ou lugares famosos, além de músicas e filmes populares (Kuppens, 2012).

Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432

Através de notícias como: “Novo vazamento de 8,4 bilhões de senhas pode ser o maior da história” (CNN, 2021), “246,6 milhões de contas de brasileiros já vazaram na internet” (Canaltech, 2022) é visível que a base de dados para a criação de dicionários é imensa.

2. METODOLOGIA

Conta com uma pesquisa bibliográfica, seguida de testes experimentais levando em conta dados estatísticos. Contextualizando o tema e os conceitos para em seguida realizar os testes propostos. A pesquisa bibliográfica foi realizada principalmente em livros e artigos de especialistas e pesquisadores e em instituições reguladoras como a NIST.

Os testes foram realizados em um computador com um processador Intel Core i7-8700, 16Gb de memória RAM 2400Mhz e o mais importante para os testes que utilizam amplamente do poder da placa de vídeo: GTX 1060 com 3GB de memória utilizando o sistema operacional Ubuntu 22.10. O aplicativo utilizado para os testes foi o *Hashcat v6.2.6* que possui código aberto. Os testes foram feitos com a função hash SHA1, para conseguir resultados mesmo sem muito poder computacional. E também um teste com a função BCrypt para comparar com o SHA1.

3. ENTROPIA

As senhas sempre estiveram associadas a segurança de alguma forma. Diversas vezes a literatura mostra como elas podem servir para distinguir amigos de inimigos, da mesma forma que pode abrir diversas portas (Burnett, 2006).

Uma senha pode ter diversas características, uma delas, por exemplo, é a entropia. O conceito de entropia foi introduzido por Claude Shannon em 1948 em seu artigo "A Mathematical Theory of Communication" como uma medida de incerteza associada a uma variável aleatória, e também conhecido como “Shannon Entropy”.

(Burr et al, 2006) mostra que Shannon estava interessado em descobrir quantos bits seriam necessários para codificar sequências de caracteres do inglês comum da maneira mais eficiente possível. Desde que Shannon cunhou o termo “entropia”, ele tem sido usado na criptografia como uma medida da dificuldade em adivinhar ou determinar uma senha ou chave.

Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432

(Yazdi, 2011) define entropia de Shannon de forma matemática sendo x uma variável aleatória dentro de um alfabeto possível de caracteres, a entropia e aleatoriedade da variável x é:

$$H(x) = -\sum_{x \in X} p(x) \log_b \quad P(x) \quad (1)$$

Por exemplo, jogando uma moeda justa, a entropia resulta em um bit, já que a chance de cair cara ou coroa é de 50%.

Para senhas escolhidas aleatoriamente, é possível calcular a entropia através da seguinte fórmula, que equivale a outros métodos utilizados para esse fim, por exemplo a entropia de Shannon:

$$H = \log_2(b^l) \ (2)$$

Então b seria o número de caracteres possíveis, l por sua vez é o número de caracteres que foram utilizados e H é o número da entropia calculada em bits. Nesse caso os valores de b são variáveis de acordo com o conjunto de caracteres que será permitido, então todos os caracteres ASCII que são imprimíveis totalizam 95.

De acordo com o NIST (National Institute of Standards and Technology) em *Electronic Authentication Guideline SP-800-63* (Burr et al, 2006) a entropia pode ser estimada de duas formas, sendo a primeira utilizando a fórmula citada anteriormente caso a senha seja escolhida de forma aleatória. Caso a senha for escolhida pelo usuário a entropia final é a somatória de:

- 4 bits no primeiro caracter.
- Do segundo ao oitavo é de 2 bits por caracter.
- Do nono ao vigésimo é de 1.5 bits por caracter.
- Após o vigésimo primeiro é de 1 bit.
- Mais 6 bits caso tenham letras maiúsculas e números ou símbolos.
- Mais 6 bits caso não esteja presente em nenhum dicionário de senhas.

Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432

4. MÉTRICAS

Como Rubens Douglas Roccia afirma em sua pesquisa, uma senha que tem um alto nível de incerteza pode ser resistente a ataques de força bruta simples, que só seguirão combinações simples, porém nos dias de hoje devido ao avanço da computação, isso é ineficaz, já que aplicativos que realizam quebra de senhas fazem o uso de dicionário, que reúne combinações comuns entre usuários (Roccia, 2021).

Mesmo a entropia sendo uma característica muito importante e aceita pela comunidade, ela por si só não é a melhor maneira de avaliar a segurança de uma senha (Weir et al., 2010).

(Kissell, 2018) prova que aumentar o tamanho da senha é mais efetivo que aumentar o conjunto de caracteres, considerando uma senha de 10 caracteres é possível verificar através da Eq. 2 que caso o conjunto de caracteres seja 95, temos o valor de entropia de aproximadamente 65.69, porém caso aumentássemos o número de caracteres em apenas um, esse valor seria em média 72.26, por outro lado, mantendo o tamanho em 10 e um conjunto de caracteres de 100, teríamos um valor de entropia próximo de 66.43.

4.1 EXEMPLOS DE SENHAS CONSIDERADAS FORTES E FRACAS

(Kuppens, 2012) destaca em seu trabalho que a criação de senhas usando a técnica mnemônica torna possível a memorização de senhas mais longas, após retratar que uma das dificuldades na atualidade é fazer com que o usuário não repita as senhas e que todas elas tenham um tamanho considerável. Utilizando essa técnica combinada a uma entropia alta e um número de caracteres sempre acima de 18 para ser considerada uma senha forte, podemos estabelecer essa classificação:

Tabela 1 - Comparação entre senhas consideradas fracas e fortes.

Senhas consideradas fracas	Senhas consideradas forte
cachorro123	m3uC4ch0rr0é!ncr1VL@
Lucas23042002	Ch1c0SoRV4Z!&F1lm3
Eamhme3	3GmTd5up3rn4T!<34R*
ACDCbb4ev	#Ur!4rT3Be5T!d10EmP3sQz$

Fonte: Elaborado pelos autores (2022).

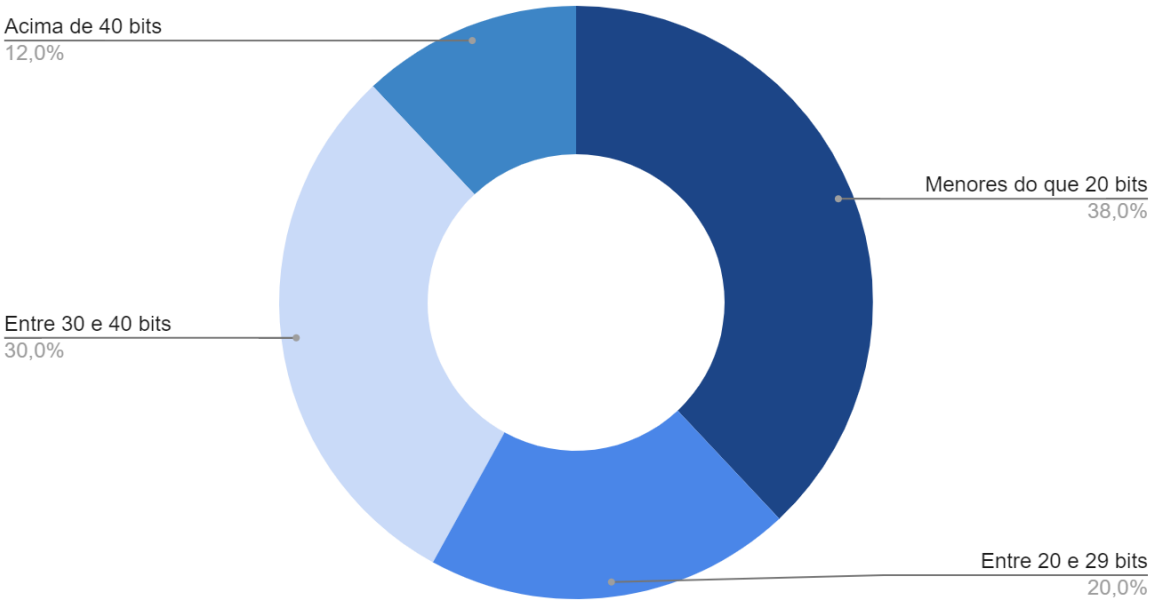
Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432

5. VAZAMENTOS

Através do portal de notícias (Tribuna Online, 2022) vemos por exemplo que o Meta, empresa dona do Facebook e entre outras plataformas, divulgou um vazamento de um milhão de senhas. Essas notícias são comuns e aparecem frequentemente no dia a dia, mostrando que é uma ameaça real.

De acordo com o (G1, 2021), através de um levantamento de senhas feito pelo *NordPass* é possível ver que a senha "123456" foi encontrada mais de 1 milhão de vezes em vazamentos no Brasil. Essa matéria também fornece as 50 senhas mais vazadas no Brasil, as quais as 20 primeiras serão usadas para verificar quanto de entropia elas possuem.

Figura 1 - Gráfico relacionando entropia e as senhas



Fonte: Elaborado pelos autores (2022).

6. ARMAZENAMENTO DE SENHAS

Uma das principais formas de armazenamento de senhas em banco de dados é através do uso de funções *hash*, que consistem em funções que transformam uma senha comum em formato de texto para um conjunto de números e letras que só podem ser gerados

exclusivamente pelo texto informado, ou seja, cada *hash* tem a proposta de ser único e exclusivo de cada conjunto de caracteres ou senha.

Segundo o NIST, um *hash* deve possuir duas propriedades: ser computacionalmente inviável de encontrar uma entrada através da saída criptografada e ser impossível de encontrar duas entradas que geram uma mesma saída, não podendo haver colisão de *hash*. (NIST, 2019)

Existem algoritmos de *hash* que possuem mais segurança do que outros, isso devido à complexidade nos cálculos realizados para mascarar a palavra, entretanto essa segurança adicional vem acompanhada de um custo computacional maior para realizar o *hashing*. Então essas funções podem ser divididas em *fast hash*, quando o *hash* é feito com baixo custo computacional e *slow hash*, quando a velocidade é reduzida, favorecendo a complexidade e qualidade do resultado.

Os testes de força bruta a seguir foram realizados com senhas que foram transformadas pelos algoritmos MD5, Sha1 e BCrypt. Sendo o BCrypt mais demorado que os outros citados e por consequência o que apresentou maior segurança e maior demora para ser testado, mesmo apresentando uma entropia menor do que nos testes realizados com outras funções.

7. ATAQUES DE FORÇA BRUTA

Uma senha pode ser quebrada de diversas formas: programas maliciosos como *keyloggers* e *screenlogger* que monitoram o que o alvo digita e visualiza em seu computador, engenharia social, força bruta, dicionário, entre outros (Kissell, 2018).

O ataque de força bruta consiste em esgotar as possibilidades de senhas, utilizando de poder de processamento para tentar todas as opções possíveis de todos os caracteres, esse tipo de ataque demanda muito tempo e capacidade de equipamento. Esses ataques podem ser realizados de duas formas: *online* e *offline*. A diferença entre um ataque *online* e um *offline* e que no online a segurança do sistema onde as senhas estão armazenadas é um fator a se levar em conta, enquanto no *offline* as senhas já estão na posse do atacante, mas em formato de *hash* (Burnett, 2006).

Existem diversos softwares de código aberto que possuem a finalidade de executarem ataques de força bruta, para isso a aplicação testa utilizando principalmente o processamento da GPU nas possíveis senhas. O teste consiste em selecionar um conjunto de caracteres, utilizar a função *hash* e comparar com o *hash* que está tentando descobrir. Esse processo pode verificar,

Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432

no computador utilizado para os testes, 11 bilhões de hash por segundo do tipo MD5, 3 bilhões por segundo do tipo SHA1, mas o número se torna bem menor conforme aumenta a complexidade do algoritmo: no BCrypt podem ser testados 8801 senhas por segundo.

7.1 POSSIBILIDADES DE SENHAS

Um dos principais pontos que dificultam o ataque de força bruta são as diversas possibilidades de símbolos que podem ser inseridos nas senhas. No alfabeto, são 26 letras considerando maiúsculas e minúsculas chegamos em um número de 52 letras no total. Os números no sistema decimal vão de 0 a 9, logo outras 10 possibilidades. Somando as possibilidades anteriores com os outros símbolos no teclado ABNT totalizam 95 opções de caracteres para cada unidade de tamanho da senha, ou seja 95^n possibilidades, sendo n o tamanho da senha.

Para contornar o grande número de aleatoriedades, os programas que realizam os ataques de força bruta, inclusive o *Hashcat* que foi utilizado nos testes permitem a utilização de "máscaras", definindo um conjunto de caracteres baseados para uma posição, diminuindo assim a quantidade a ser testada. Essas máscaras podem ser personalizadas e são baseadas em padrões comportamentais e em vazamentos anteriores de senhas, como começar com letra maiúscula e o restante números, por exemplo. Nesse formato, senhas de 8 dígitos podem ser quebradas em menos de 9 segundos, conforme a tabela a seguir.

Tabela 2 - Entropia de senhas e tempo para serem quebradas, algoritmo SHA1.

Senha	Entropia	Tempo do teste de força bruta	Tempo do teste de máscara
9635	13 bits	12 segundos	1 segundo
96358	16 bits	9 segundos	1 segundo
P457	20 bits	5 segundos	1 segundo
Pq23	23 bits	7 segundos	1 segundo
96478521	26 bits	18 dias	9 segundos
Pq23%	30 bits	6 segundos	2 segundos

Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432

Senha	Entropia	Tempo do teste de força bruta	Tempo do teste de máscara
Pq23%\	39 bits	169 segundos	179 segundos
Henrique	45 bits	18 dias	1 segundo
Jp5874!/	52 bits	18 dias	154 segundos
Lu20221611!	67 bits	Mais de uma década	13 segundos

Fonte: Elaborado pelos autores (2022).

Observando os dados da tabela acima, é possível analisar que mesmo senhas com entropia maior do que outras, como “96358” e “Pq23%” que possuem 14 bits de diferença, não tem uma diferença de tempo de quebra considerável.

Outro ponto a ser identificado é que mesmo senhas que apresentam uma entropia considerável como “Lu20221611!” que possui 67 bits, pode ser facilmente descoberta utilizando máscaras padrões dos programas de quebra de senha, uma vez que o padrão de construção dessa senha é comum. Entretanto demoraria décadas para descobrir utilizando somente força bruta.

7.2 A RELAÇÃO DO TEMPO DE QUEBRA DA SENHA COM SUA ENTROPIA

Considerando a fórmula de entropia discutida anteriormente, um determinado tamanho de senha possui uma quantidade limitada de entropia que pode ser atingida, então é possível estimar a quantidade de tempo que levaria para a máquina utilizando o aplicativo “hashcat” descobrir todas as senhas de um conjunto de valores de entropia.

Tabela 3 - Quantidade de dígitos da senha, tempo estimado para quebra e entropia

Dígitos da senha	Tempo	Entropia
4	1 segundo	17 a 25 bits
5	3 segundos	26 a 32 bits
6	3 minutos	33 a 39 bits

7	5 horas	40 a 45 bits
8	18 dias	46 a 52 bits
9	5 anos	53 a 59 bits

Fonte: Elaborado pelos autores (2022).

Para uma senha ter uma quantidade de proteção puramente baseada em entropia, são necessárias mais do que 53 bits, para que a senha demore aproximadamente 5 anos para ser descoberta caso ocorra um vazamento. Entretanto somente a entropia não é garantia de segurança. Se a senha foi encontrada em outro vazamento ou possui um padrão comum de construção, esse valor passa a ser insignificante.

7.3 TESTES COM A FUNÇÃO HASH BCRYPT

Para verificar o quanto a tecnologia de armazenamento das senhas importam, foi testada a senha “Carro”, essa senha possui apenas 28 *bits* de entropia, e é considerada uma senha que não apresenta uma segurança elevada pois, se estiver armazenada através de um *fast hash* como SHA1 e atacada com algoritmo de força bruta e quebrada em menos de três segundo. Porém, ao testar a mesma senha com outra função hash, como a BCrypt, essa sendo mais complexa e requisitando maior poder computacional para funcionar, a mesma senha demorou em média 9 minutos e 57 segundos para ser descoberta. Somente com a mudança de função, a senha aumentou o tempo necessário para ser quebrada em aproximadamente 199 vezes, indicando que a entropia pode permanecer a mesma mas ainda assim a senha se tornar mais segura dependendo de como é armazenada.

8. CONSIDERAÇÕES FINAIS

Através do presente trabalho, foi possível notar que a entropia de uma senha não pode ser um fator considerado por si só, já que mesmo tendo uma entropia alta a senha pode conter diversos padrões ou informações óbvias, inclusive que já foram vazadas e estão presentes em arquivos de vazamento disponíveis livremente na internet.

Revista de Extensão e Iniciação Científica da UNISOCIESC
REIS ISSN 2358-4432

Um fator que foi notado durante os testes e também o cálculo de entropia de algumas senhas é que, o tamanho da senha influencia muito mais do que a quantidade de caracteres no conjunto que é usado para a elaboração da senha, ou seja, a entropia aumenta mais juntamente com o tempo para quebrar aquela senha.

Porém ainda existe um fator que dificulta a descoberta da senha, que seria a técnica de *hash* utilizada na senha, por exemplo, a demora para quebrar uma senha que têm o algoritmo de *hash* MD5 ou SHA1 aplicada é muito menor do que outra que tenha um algoritmo mais seguro como a função BCrypt. A razão seria que o último citado requer muito mais poder de processamento e cálculos, diferente dos primeiros.

Além de todos os métodos que podem dificultar a aplicação da força bruta, caso já tenha um alvo definido, pode se tornar mais simples encontrar a senha do mesmo, pois vimos que a aplicação de padrões e máscaras para descoberta das senhas causa um grande impacto na velocidade para encontrar todas as combinações possíveis. Construindo um dicionário que possui tudo que o alvo gosta, como filmes, músicas, além de informações pessoais, as possibilidades diminuem e naturalmente o tempo de quebra de senha também. Essas informações geralmente podem ser obtidas através de engenharia social.

Dessa forma, métricas como entropia podem ser utilizadas para a criação de uma senha segura, contudo, com os testes realizados é possível notar que só esse parâmetro não deve ser considerado, pois mesmo dentro de uma senha com um alto número de entropia, ainda podem existir padrões ou até mesmo a senha já ter sido vazada. Uma senha com entropia considerável, não seguindo nenhum tipo de padrão e com uma quantidade de caracteres acima de 11, unida a um algoritmo de *hashing* seguro e complexo demoraria muito tempo para ser quebrada.

REFERÊNCIAS

AGÊNCIA FOLHAPRESS. **Meta anuncia vazamento de um milhão de senhas do Facebook: Os perfis afetados serão notificados nos próximos dias. Tribuna Online.** 2022. Disponível em: https://tribunaonline.com.br/ciencia-e-tecnologia/meta-anuncia-vazamento-de-um-milhao-de-senhas-do-facebook-125450. Acesso em: 15 nov. 2022.

Burr W., Dodson D., & Polk W. (2006). **NIST Special Publication 800-63 Version 1.0. 2 - Electronic Authentication Guideline, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, EUA**: NIST.

BURNETT, Mark. **Perfect Passwords: Selection, Protection, Authentication. 1. ed. [S.l.]: Syngress**, 2006.

DEMARTINI, Felipe. CANALTECH. **246,6 milhões de contas de brasileiros já vazaram na internet.** Disponível em: https://canaltech.com.br/seguranca/2466-milhoes-de-contas-de-brasileiros-ja-vazaram-na-internet-219225/. Acesso em: 12 set. 2022.

G1 GLOBO. **As senhas mais comuns em vazamentos de dados no Brasil em 2021.** Disponível em: https://g1.globo.com/tecnologia/noticia/2021/11/18/as-senhas-mais-comuns-em-vazamentos-de-dados-no-brasil-em-2021.ghtml. Acesso em: 12 set. 2022.

KISSELL, Joe. **Aprendendo a Proteger suas Senhas. 1. ed. [S.l.]: Novatec Editora**, 2017.

KUPPENS, Luciano Lima. **Utilização de dicionários probabilísticos personalizados para decifrar arquivos em análises periciais.** 2012. xiii, 99 f., il. Dissertação (Mestrado em Engenharia Elétrica)—Universidade de Brasília, Brasília, 2012.

NIST. **Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography.** 2019. Disponível em: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Br2.pdf. Acesso em: 15 nov. 2022.

ROCCIA, Rubens Douglas. **Usuários respeitam as normas de criação de senhas seguras?: Uma análise de datasets de senhas vazadas.** São Paulo, 2021 Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Universidade de São Paulo Instituto de Matemática e Estatística . Disponível em: https://bcc.ime.usp.br/tccs/2020/rubensd/Monografia.pdf. Acesso em: 15 nov. 2022.

Weir M. (2010). **Using Probabilistic Techniques to Aid in Password Cracking Attacks.** Tese de Doutorado, Departamento de Ciência da Computação, Universidade do Estado da Flórida, EUA.

YAZDI , Shiva. **Analyzing Password Strength & Efficient Password Cracking. 2011**. Tese (Computer Science) - Florida State University, 2011. Disponível em: https://diginole.lib.fsu.edu/islandora/object/fsu:181989/datastream/PDF/view. Acesso em: 13 set. 2022.